PATENT
Attorney Docket No. 80738

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent No. 7,349,418

Confirmation No. 3503

Issued: March 25, 2008

Name of Patentee: Cohen

Patent Title: RESIDUE-BASED ENCODING OF PACKET LENGTHS OF PARTICULAR USE IN PROCESSING AND SCHEDULING PACKETS

REQUEST FOR CERTIFICATE OF CORRECTION OF PATENT FOR PATENT OFFICE MISTAKE (37 C.F.R. § 1.322)

Attn: Certificate of Correction Branch
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

It is requested that a Certificate of Correction be issued to correct Office mistakes found the above-identified patent. Attached hereto is a Certificate of Correction which indicates the requested correction. For your convenience, also attached are copies of selected pages (a) from the issued patent with errors highlighted, and (b) from the original application as filed July 11, 2003, with the correct text/instructions.

It is believed that there is no charge for this request because applicant or applicants were not responsible for such error, as will be apparent upon a comparison of the issued patent with

the application as filed or amended. However, the Assistant Commissioner is hereby authorized to charge any fee that may be required to Deposit Account No. 501430.

Respectfully submitted,
The Law Office of Kirk D. Williams

Date: 9-30-2009 By

Kirk D. Williams, Reg. No. 42,229
One of the Attorneys for Applicants
CUSTOMER NUMBER 26327
The Law Office of Kirk D. Williams
P.O. Box 39425, Denver, CO 80239-0425
303-282-0151 (telephone), 303-778-0748 (facsimile)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,349,418
APPLICATION NO. : 10/617,539
DATED : March 25, 2008
INVENTOR(S) : Cohen

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Col. 8, Line 23, replace "accordingly." With – accordingly *y* –

MAILING ADDRESS OF SENDER:
Kirk D. Williams, Reg. No. 42,229
Customer No. 26327
The Law Office of Kirk D. Williams
P.O. Box 39425, Denver, CO 80239

From Application filed July 11, 2003

In one embodiment, when the threshold value is not exceeded in producing the adjusted value, the value of one is added to produce the adjusted value, and the residue amount is increased by twice the absolute value of the threshold value (e.g., the threshold value is typically, but not always a negative number, so one or more mathematical operations are typically performed to actually increase the residue amount). In one embodiment, the value of *i* is added to produce the adjusted value when the threshold value is exceeded, with *i* being any integer. In this case, typically the residue amount will be increased by *i* times the threshold value. In one embodiment, the residue amount is a number maintained in a range between a maximum value *x* and a minimum value *y*, and the threshold value is accordingly *y*. In this case, typically the residue amount will be increased by *x-y+1*. These are examples where the residue amount is a forward looking indication (e.g., the residue amount indicates the cumulative adjusted values not included in the original values).

(1) see Col. 8 line 23

FIG. 1A illustrates a mechanism used in one embodiment for adjusting the identified values 100, which are received by mechanism 101, which determines and maintains adjusted values and maintains and updates one or more residue amounts. These adjusted values are typically stored in storage 102 (e.g., memory or any other storage mechanism) for use by processing element 104 (e.g., a scheduler, packet processor, or any other process or mechanism). The residue amounts used in determining the adjusted values are typically maintained in storage 103 (e.g., registers, flip-flops, memory, or other storage mechanism).

FIG. 1B illustrates a mechanism used in one embodiment for adjusting packet lengths of identified packets 109, which are received by mechanism 110. In one embodiment, mechanism 110 is an ASIC for performing queue enqueue and dequeue operations and includes a mechanism for adjusting values and maintaining and updating one or more residue amounts. As shown, mechanism 110 includes a packet length memory 111 for storing the adjusted packet lengths and a residue amount memory 112 for storing one or more residue amounts. The packets corresponding to the received packets

Thus, the adjusted packet length in the above example is roughly 1/256 of the actual packet length, but adjusted so that no matter how many packets are enqueued, the total accumulated length error over time (the residue amount) is always in the range −128 to +127. In other words, once averaged over enough packets, the accumulated error becomes negligible.

FIG. **4** illustrates pseudo-code **400** describing a process used in one embodiment. This example saves the four LSBs of the packet length, and the residue amount is limited to as small a window as possible around zero. The residue amount in this case can have sixteen possible values, and the code keeps the residue amount between −8 and +7 at all times. This ensures that the scheduler for a given queue is never off by more than −8 to +7 bytes, no matter how many packets have been served. The choice of only saving the four LSBs in the above example is arbitrary. Depending on how much instantaneous error in scheduling could be tolerated, the packet lengths could be shortened almost arbitrarily. In fact, if an instantaneous error on the order of one MTU is allowed, the packet lengths could be stored as just one bit (in which case, the residue amount would need to be the size of an MTU).

Additionally, there are many ways to encode the residue amount, and different embodiments may employ different techniques. For example, one embodiment uses a residue amount that ranges between any two numbers. For example, the residue amount may range between zero and some maximum value (e.g., instead of the range of the residue amount being centered at zero). In one embodiment, the adjusted packet lengths are determined in another manner rather than using a simple "round up" or "round down" (e.g., truncated) decision based on the residue amount, and could in fact adjust the stored packet length by any values or amounts. Moreover, one or more of these adjustment amounts may vary over time, such as in response to threshold values being exceeded.

Additionally, the techniques described herein are extensible to many different applications, and are not limited to adjusting packet lengths. For example, these techniques can be used for dynamically adjusting values (e.g., a packet length, processing amount, a size indication, or any other value) corresponding to items of a stream of items or items associated with any entity based on one or more maintained residue amounts in a manner typically to reduce the long-term effect of the error induced by using the adjusted values rather than the original values. Also, the number of embodiments and the manner in which the values are adjusted and the one or more residue amounts are updated are extensible in keeping within the scope and spirit of the invention.

For example, in one embodiment, a threshold value is used in determining how to adjust a particular value. When the threshold value is exceeded in producing an updated residue amount, the value of one is added to produce the adjusted value, and the residue amount is decreased by twice the threshold value. Different embodiments may use different threshold values, which are typically selected to match the needs of the application. In one embodiment, the threshold value is a power of two, and thus, two times the threshold value (e.g., using a multiplication, addition or shift operation, etc.) is the next larger power of two. In one embodiment, an actual mathematical operation (e.g., addition or subtraction) is performed to reduce the residue amount. In one embodiment, overflow bits (i.e., those greater than that required to store the maximum value of the residue) are ignored in producing the updated residue amount. In one embodiment, the value of i is added to produce the adjusted value when the threshold value is exceeded, with i being any integer. In one embodiment, the residue amount is a number maintained in a range between a maximum value x and a minimum value y, and the threshold value is accordingly x+1. In this case, typically the residue amount will be decreased by x−y+1. These are examples of embodiments where the residue amount is a lagging indication (e.g., the residue amount indicates the cumulative original values not included in the adjusted values).

In one embodiment, when the threshold value is not exceeded in producing the adjusted value, the value of one is added to produce the adjusted value, and the residue amount is increased by twice the absolute value of the threshold value (e.g., the threshold value is typically, but not always a negative number, so one or more mathematical operations are typically performed to actually increase the residue amount). In one embodiment, the value of i is added to produce the adjusted value when the threshold value is exceeded, with i being any integer. In this case, typically the residue amount will be increased by i times the threshold value. In one embodiment, the residue amount is a number maintained in a range between a maximum value x and a minimum value y, and the threshold value is accordingly. In this case, typically the residue amount will be increased by x−y+1. These are examples where the residue amount is a forward looking indication (e.g., the residue amount indicates the cumulative adjusted values not included in the original values).

(1) should be "accordingly y."

FIG. **1A** illustrates a mechanism used in one embodiment for adjusting the identified values **100**, which are received by mechanism **101**, which determines and maintains adjusted values and maintains and updates one or more residue amounts. These adjusted values are typically stored in storage **102** (e.g., memory or any other storage mechanism) for use by processing element **104** (e.g., a scheduler, packet processor, or any other process or mechanism). The residue amounts used in determining the adjusted values are typically maintained in storage **103** (e.g., registers, flip-flops, memory, or other storage mechanism).

FIG. **1B** illustrates a mechanism used in one embodiment for adjusting packet lengths of identified packets **109**, which are received by mechanism **110**. In one embodiment, mechanism **110** is an ASIC for performing queue enqueue and dequeue operations and includes a mechanism for adjusting values and maintaining and updating one or more residue amounts. As shown, mechanism **110** includes a packet length memory **111** for storing the adjusted packet lengths and a residue amount memory **112** for storing one or more residue amounts. The packets corresponding to the received packets **109** are stored in packet storage **114** (e.g., DRAM or other memory or storage mechanism).

Scheduler **116** receives the adjusted packet lengths **115** (or values based thereon such as a sum of multiple adjusted packet lengths). Based on the adjusted packet lengths **115** and its scheduling mechanism, scheduler **116** determines the scheduled queue indications **117**. In response, packet processing engine **120** sends requests **121** to mechanism **110**, which retrieves the corresponding packets from packet storage **114** and transmits them to packet processing engine **120** as indicated by packets **122**. In one embodiment, packet processing engine **120** includes scheduler **116**.

FIG. **1C** illustrates one embodiment for adjusting values. One or more interfaces, processes, threads, information streams, etc. **130** generate packets or other items **131-139** associated with values, which are typically stored in queues **141-149**. Queues **141-149** include one or more mechanisms to adjust these associated values and maintain and update the